

---

title: Pandocチュートリアル 第1回 Pandocでドキュメントを変換しよう

author: 藤原 惟 / すかいゆき (@sky\_y)

date: 2017年1月20日

revealjs-url: reveal.js-3.4.0

theme: sky-sky-y

transition: fade

transitionSpeed: fast

slideNumber: true

margin: 0

...

---

## 自己紹介

---

- 名前
    - 藤原 惟
    - すかいゆき
    - Yuki Fujiwara (本名)
  - 職業
    - フリープログラマ
    - 専門学校 非常勤講師
- 

## Pandocに関する活動

---

- Qiitaを中心に記事執筆
    - [多様なフォーマットに対応！ドキュメント変換ツールPandocを知ろう - Qiita](#)
  - Pandocユーザーズガイドを和訳
    - [Pandoc ユーザーズガイド 日本語版](#)
    - バージョンが古くなつたので、改訂を予定
- 

## 発表を始めます

---

# Pandoc公式サイト

---

- [Pandoc - About pandoc](#)
  - ユーザーズガイド
    - [Pandoc - Pandoc User's Guide](#)
    - [Pandoc ユーザーズガイド 日本語版](#)
- 

## Pandocと私

---

- 院生時代の依頼
    - 従来よりLaTeXで書かれた「教育用システムの利用手引」を、EPUBにできるかどうか検証してくれないか？
    - このとき調べ回って、「Pandocというものがあるらしい」と気づいた
  - 「LaTeXからEPUBに変換する」タスクが想像以上にあっさり実現
    - 少少は難はあったが（input命令に当時は対応していない（今は対応）とか、組版がいけてないとか）
    - この感動が、今の活動の原点
  - なんでこんなにすごいソフトウェアが、日本では知られていなかったんだろう？
    - 「Pandocを広めよう」と決意した瞬間
- 

## このチュートリアルでやること（2回分の概要）

---

- 第1回（今回）Pandocでドキュメントを変換しよう
    - Pandocをツールとして使う（入門編、第2回の基礎知識）
  - 第2回 HaskellでPandocを拡張してみよう
    - Haskellのやさしい入門（を目指します）
    - 「日常的な実用言語」としてのHaskellを体験してもらいたい
    - Pandocのソースコードも少し読みます
-

# Markdownって何？

---

- このスライド自体が、実はMarkdownで書かれています
  - 元々は[John Gruberが作ったオリジナルの処理系](#)でHTMLに変換するための略記法だった
  - そのうちGitHubやPHPなどで記法が拡張された
    - MultiMarkdownやPandocの登場をきっかけに、目的も「論文」「プレゼンーション」「電子書籍」など用途が広がった
    - 数々の「方言」がある状態
  - しかし、本当に基本のMarkdownだけを覚えれば、大抵は書けます
    - プレビューを行うのが鉄則
- 

## こんなことに困っていましたか？

---

- 書類をWord形式で提出しなければならないけど、Word重いし面倒
  - バージョン管理をしたいけど、Word文書は `git diff` とか取りにくい
  - Word文書をHTMLに変換せよとお達しがあった
  - 手元にLaTeXのソースがあって、それをEPUBにしてくれと言わされた（実話）
  - 卒論のLaTeX辛いので、Markdownで書きたい
  - Markdownでスライドショーを作りたい
- 

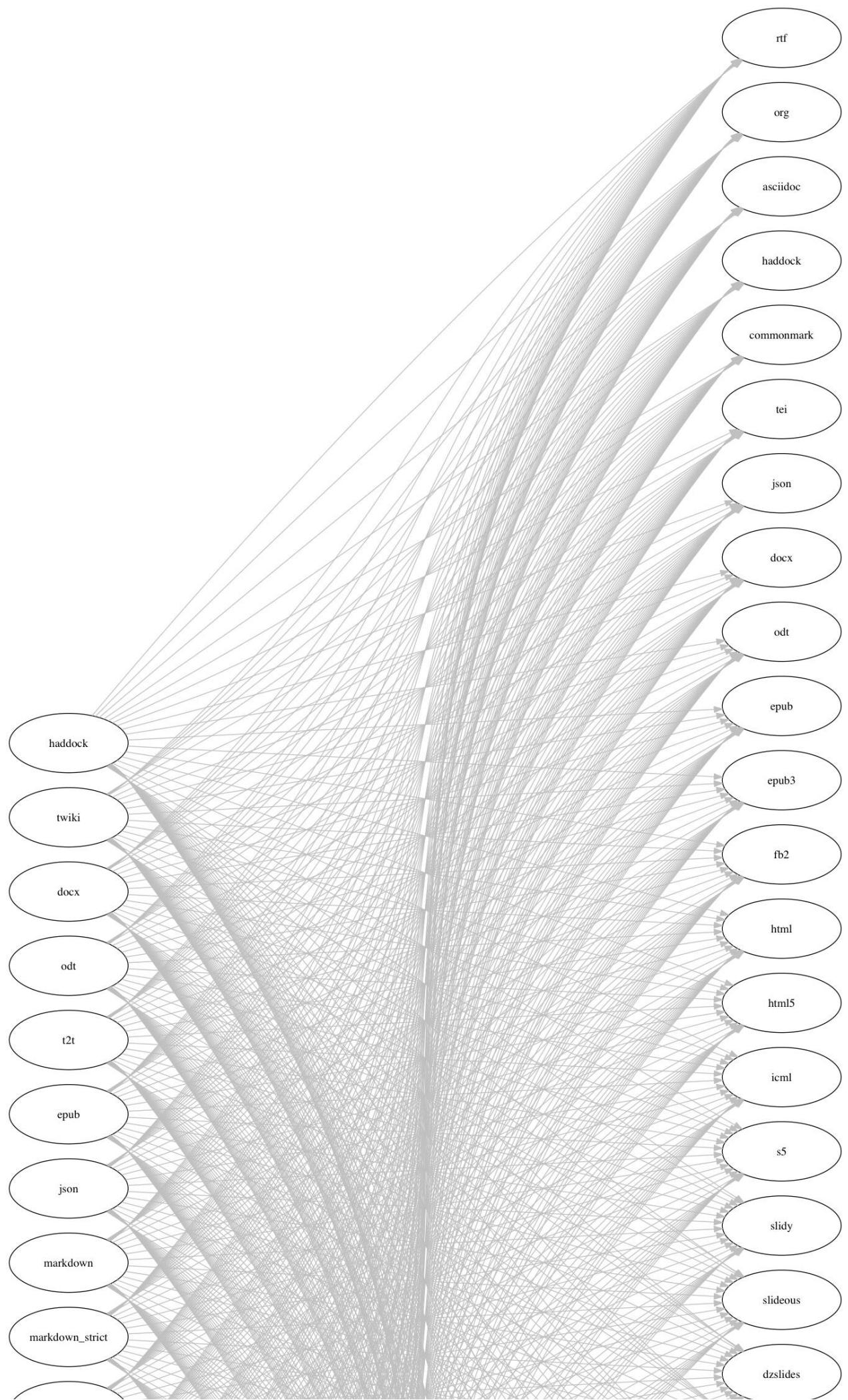
## Pandocの魅力

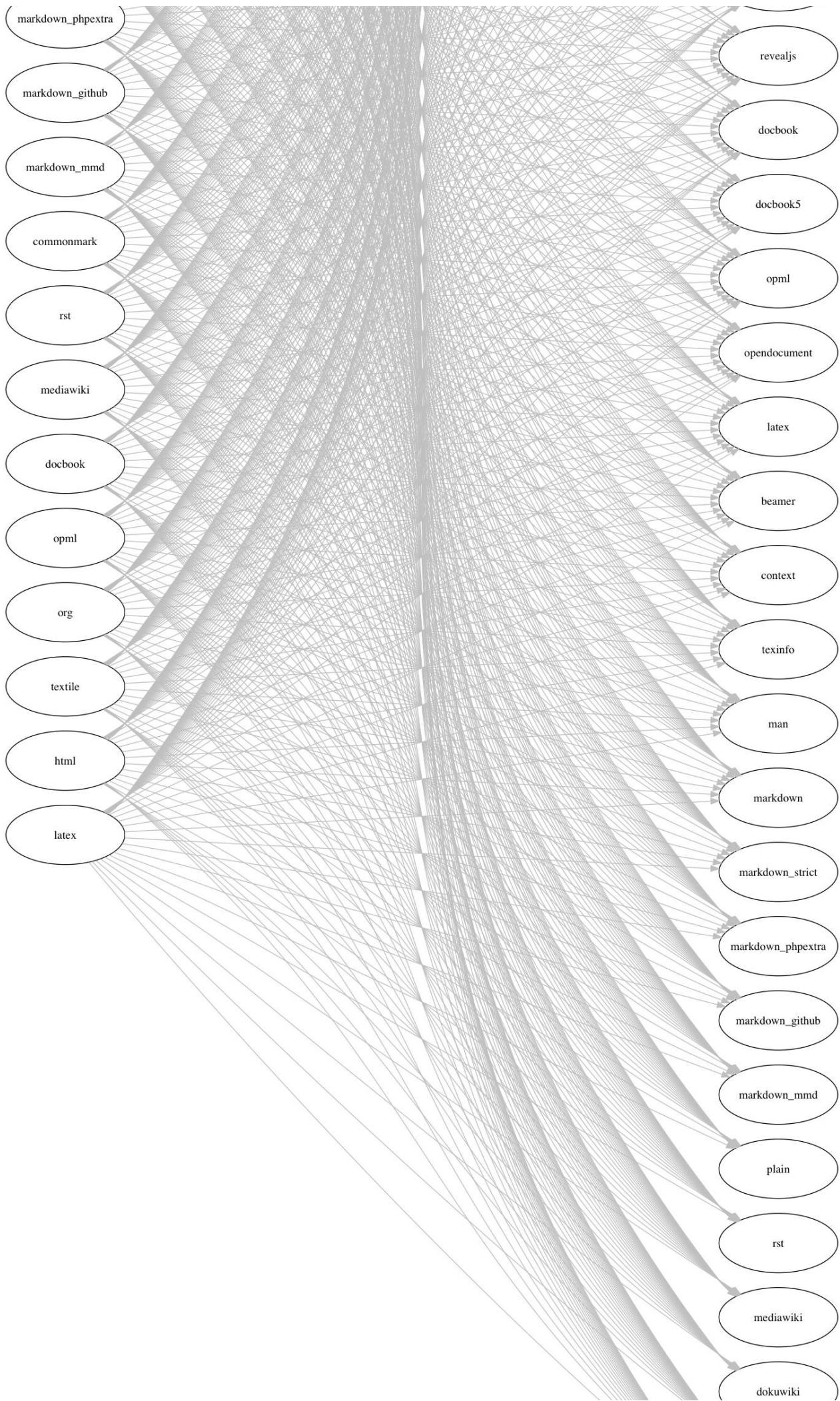
---

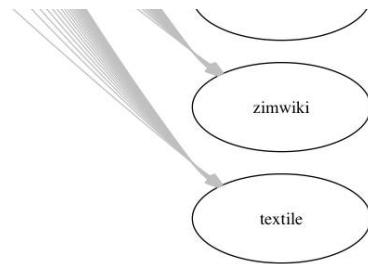
## Pandocとは

---

- [Pandoc - About pandoc](#)
  - 文書変換ツール
    - あるフォーマットで書かれた文書を、別のフォーマットに変換するツール
  - Pandocの特徴は、対応フォーマットが非常に多いこと
-







{ width=15% }

---

## 対応フォーマット（一部省略）

---

- 入力
    - Markdown (Pandoc, CommonMark, PHP Markdown Extra, GitHub-Flavored Markdown, MultiMarkdown)
    - (subsets of) Textile, reStructuredText, HTML, LaTeX, MediaWiki markup, Emacs Org mode
    - OPML, DocBook, EPUB, ODT and Word docx
  - 出力
    - Markdown (同上)
    - manページ, AsciiDoc, InDesign ICML
    - プрезентーション: LaTeX Beamer, HTML5(reveal.jsなど)
    - PDF (wkhtmltopdfまたはLaTeXエンジンが必要)
- 

## Pandocを使う心得

---

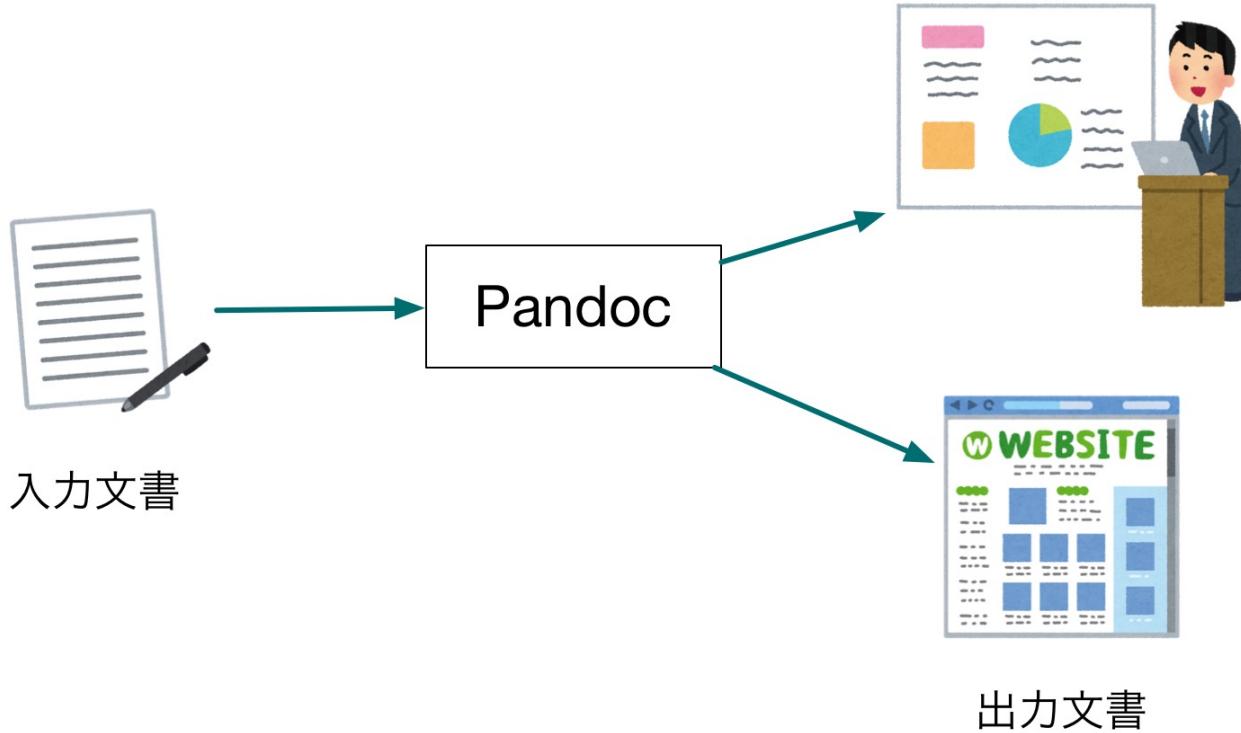
- 過剰な期待をし過ぎないこと
    - Pandocは万能でないし、文書仕様の全てを満たしているわけではない
  - 補助的に使うのがベスト
    - Pandocで、テキストと大まかな構造を抽出
    - 変換し切れなかった部分を、手作業や自作スクリプトで編集
- 

## Pandocの実装

---

- 言語: Haskell
  - Pandoc的には、「厳密に型が定義されている」ことがありがたい

- Haskellは構文解析器(パーサ)を作るのにすごく適している (Parsecなど)
- モジュール構成
  - Reader: 入力文書を解析し、Haskell上の中間文書に変換する
  - Writer: 中間文書を受け取り、出力フォーマットに変換する



## Pandocが扱えるMarkdown方言

- Pandoc's Markdown: `-f markdown`
  - Pandocにおける標準のMarkdown方言
  - 技術文書から論文・電子書籍まで幅広く対応
- GitHub Flavored Markdown (gfm): `-f markdown_github`
  - プログラマ・フレンドリーな方言
- PHP Markdown Extra: `-f markdown_phpextra`
  - 最近はMarkdown Extraとも呼ばれる
- MultiMarkdown: `-f markdown_mmd`
  - HTMLだけでなくLaTeXなどの論文も意図した処理系
- CommonMark: `-f commonmark`

# 準備: Pandocをインストールする

---

## ターミナルを開く

---

- Mac: ターミナル.app or iTerm2
  - Windows: (今回は)コマンドプロンプト
    - (分かっている方は)お好きなターミナルでも結構です
- 

## Pandocのインストール

---

- Haskell処理系は不要です
  - パッケージを直接落としてインストール
    1. [ここからパッケージをダウンロード](#)
      - Windows: `.msi`, Mac: `.pkg`
    2. インストール
  - パッケージマネージャでインストール
    - Mac([Homebrew](#)): `$ brew install pandoc`
    - Windows([Chocolatey](#)): `> cinst pandoc`
      - Chocolateyのみ、PowerShell（管理者権限付き）がおすすめ
    - Linux(Debian): `$ sudo apt-get install pandoc`
- 

## wkhtmltopdfのインストール

---

- パッケージを直接落としてインストール
    1. [ここからパッケージをダウンロード](#)
      - Windowsは未検証ですが、MinGWの方を試してみてください
    2. インストール
  - パッケージマネージャでインストール
    - Mac: `$ brew cask install wkhtmltopdf`
      - Caskの方なので注意
-

# 動作確認: Pandoc単体

---

Bashにて確認（コマンドプロンプトも同様のはず）

```
$ pandoc --version  
$ pandoc --list-input-formats  
$ pandoc --list-output-formats  
$ echo "##Hello##" | pandoc -f markdown -t html  
<p><strong>Hello</strong></p>
```

---

# 動作確認: Pandoc + wkhtmltopdf (PDF)

---

```
$ echo "##Hello##" | pandoc -f markdown -t html -o hello.pdf
```

---

## このスライドを自分で作ろう

---

## Pandocで作れるスライド

---

- 今回は「reveal.js」形式に変換
  - HTML+JavaScriptによるプレゼンテーション
- 補足: Pandocでは他のプレゼン形式にも変換できる
  - LaTeX Beamer
  - reveal.js以外のHTMLプレゼン（割愛）

---

## reveal.js

---

- HTML/CSS/JavaScriptで実装されたプレゼンWebアプリ

- クライアントサイドで完結→GitHub Pagesにアップロード可能
- 

## 実際のソースコード

---

- このスライド自体はGitHub Pagesでアップされています
- GitHubリポジトリ: <https://github.com/sky-y/haskell-skype-pandoc-1>
  - Markdown (raw): <https://raw.githubusercontent.com/sky-y/haskell-skype-pandoc-1/master/index.p.md>

```
$ git clone https://github.com/sky-y/haskell-skype-pandoc-1.git`
```

## reveal.jsのライブラリを入れる

---

- 本来の手順
  - [Releases · hakimel/reveal.js](#) の 「Downloads」 にあるzipまたはtar.gzをダウンロードし、展開する
  - フォルダの中の `index.html` を書き換えて、ダブルクリックするとそのままブラウザで動く
- 今回
  - テーマ(CSS)をカスタマイズしている
    - このスライドでは `sky` をカスタマイズしたCSS( `sky-sky-y` )を作った
    - 上記のGitHubリポジトリの 「reveal.js-3.4.0/css/theme」 に置いている
  - `$ cd haskell-skype-pandoc-1/`

---

## Markdownを書く

---

- 今回はタイトルを編集してみましょう
  - 2行目: `title:` (ここを変更する)
  - 3行目: `author:` (ここを変更する)
- その他は、PandocのMarkdownで書く
  - [Pandoc - Pandoc User's Guide](#)
    - [Pandoc ユーザーズガイド 日本語版](#)

# 補足: ヘッダについて

---

- ヘッダには2種類ある
    - Title block (% ではじまる、簡潔)
    - YAML metadata ( --- ではじまり ... でおわる、高機能)
      - テンプレート内で使用するための変数を埋め込める
  - 詳細はユーザーズガイドの「[Producing slide shows with pandoc](#)」を参照
- 

```
---
```

```
title: ほげ
author: (あなたの名前)
date: 2017年1月20日
revealjs-url: reveal.js-3.4.0
theme: sky-sky-y
transition: fade
(略)
...
```

## pandocコマンド

---

```
$ pandoc index.p.md -s -t revealjs -o index.html
```

- `-s` : standalone (ヘッダ付きの完全なファイルを出力)
  - `-t` : 出力フォーマット(reveal.js)
  - `-o` : 出力ファイル名
- 

## プレゼンファイルを開く

---

```
$ open index.html    # Mac/Linux
> start index.html  # Windows
```

# HTMLを変換してみる

---

## 例: connpassのHTMLからMarkdownを求める

---

- 準備の際に実際にやりました
  - オーガナイザーさんにconnpassページ原稿を渡したい
  - やっぱりMarkdownで書きたい
    - connpass自体はMarkdown記法を持っている
  - 藤原はconnpassの本イベントページを編集する権限がない
    - そうだ、HTMLをPandocで読み込めば、Markdownがゲットできる！！
- 

## connpassのHTMLを取得

---

- Pandocチュートリアル 第1回 - connpass
  - メニューアイコン（または右クリック）で「ソースを表示」
  - そのまま Command-S (Ctrl-C) でHTMLを保存( connpass.html とする)
- 

## Q: このMarkdownは何の方言に近いか？

---

- connpassで使えるMarkdown記法 - connpass ご利用ガイド
  - ヒント
    - `http://connpass.com` URLは自動的にリンクになります。
    - `http://connpass.com` URLは自動的にリンクになります。
-

# A: GitHub Flavored Markdown(gfm)に近い

---

- Pandocのデフォルトで使えるMarkdown方言のうち、自動リンク記法はgfmが対応
  - このように、出所の分からないMarkdown仕様でも、ある程度なら処理系を推測できる
    - (今後の課題として、メジャーなサービスについて方言を比較できるようにしたい)
- 

## PandocでMarkdownに変換

---

```
$ pandoc connpass.html -t markdown_github -o connpass.md
```

---

## LaTeXを変換してみる

---

## LaTeXを変換してみる

---

- 注意: 文字コードはUTF-8に統一してください
  - サンプル
    - [TeX入門](#)
    - \$ cd .. /example-paper
    - `input.tex` というファイル
- 

## LaTeXを変換してみる

---

```
$ pandoc input.tex -s -o output.docx
```

---

# スタイルを変更する (割愛)

- 参考: [プログラマの文書作成術 : Markdown, LaTeXなどの文書からWord文書を生成する - Qiita](#)

## 質問・作業・もくもく会

- Slackにて受け付けます (#field)
- TwitterでもOKです
  - @sky\_y | Twitter

## 補足: Pandocの今後の課題

- 日本語に特化した文書フォーマットにほとんど対応していない
  - 書籍におけるルビや圈点など
  - 日本語コミュニティの必要性
- 表形式の文書は対応していない
  - Excel文書など→Excel方眼紙への対策には致命的
  - サードパーティのプリプロセッサにより部分的に変換する手段はある
    - 一部の図表 (Graphvizなど) はこの方法で取り込むことができる